10

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/626,340	07/24/2003	Heng Chu	RSW920030074US1	9120

25259 7590 10/04/2006

IBM CORPORATION
3039 CORNWALLIS RD.
DEPT. T81 / B503, PO BOX 12195
REASEARCH TRIANGLE PARK, NC 27709

EXAMINER	
LOVEL, KIMBERLY M	
ART UNIT	**PAPER NUMBER**
2167	

DATE MAILED: 10/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/626,340	**Applicant(s)** CHU ET AL.	
	Examiner Kimberly Lovel	**Art Unit** 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 20 July 2006.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-31 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-31 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 20 July 2006 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This communication is responsive to the Amendment filed 20 July 2006.

2. Claims 1-31 are pending in this application. Claims 1, 13, 24, 26 and 31 are independent. In the Amendment filed 20 July 2006, claims 1, 2, 4, 5-8, 10, 11, 13, 17, 20-26 and 31 have been amended. This action is made Non-Final.

3. The rejections of claims 1-12, 24-26 and 29-30 as being anticipated by US PGPub 2003/0208498 to Feinberg et al; claims 13-23 and 27-28 as being unpatentable over US PGPub 2003/0208498 to Feinberg et al in view of the article "XQuery 1.0 and XPath 2.0 Functions and Operators" written by W3C; claim 31 as being unpatentable over US PGPub 2003/0208498 to Feinberg et al in view of US Patent No 6,996,589 to Jayaram et al have been withdrawn based on applicants' arguments and amendments.

Drawings

3. The objections to the drawings have been withdrawn as necessitated by the amendment.

Claim Objections

4. Claims 13 and 24 are objected to because each preamble states the terminology "further comprising steps of." Since the claim is an independent claim, it is suggested that the applicants remove the word "further," since if a claim does not already comprise of steps, it is unclear how the claim can *further* comprise of additional steps. Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The rejections of claims 1 and 13 under 35 U.S.C. 112, first paragraph, as representing single means claims are withdrawn as necessitated by the amendment.

6. The rejections claims 13 and 24 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention are withdrawn since the rejection has been replaced by an objection to the claims.

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 1 is rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01. The omitted step is the validation of the input. The claim states "the generated output adheres to a different syntax level than a syntax level used when validating the input." Since the step of validating the input has not previously occurred, it is unclear the meaning of the limitation.

To allow for compact prosecution, the examiner will apply prior art to these claims as best understood, with the assumption that applicant will amend to overcome the stated 112 rejections.

Claim Rejections - 35 USC § 101

8. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

9. The rejections of claims 1-23 under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter are withdrawn as necessitated by amendment.

10. Claim 24 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 24 recites a system for applying abstraction to object markup definitions, further comprising: a validating parser usable by a computer; first means for using the validating parser to validate an input document expressed as an object markup definition, wherein the validation is performed according to a syntax level which allows the object markup definition to be successfully validated; and second means for using the validating parser to apply abstraction to the object markup definition when generating at least one output object for use by a computer application therefrom, responsive to the first means, wherein the applying of the abstraction generates at least one output object according to a different syntax level which would not allow the object markup definition to be successfully validated.

Even though claim 24 recites a system, according to the applicants' specification, the system can be entirely embodied in software. According to MPEP 2106:

> The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a

> composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se*.
>
> Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)
>
> Merely claiming nonfunctional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because "[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.").

Claim 25 is dependent on the system of claim 24, and therefore is rejected on the same grounds as claim 24.

To allow for compact prosecution, the examiner will apply prior art to these claims as best understood, with the assumption that applicant will amend to overcome the stated 101 rejections.

Claim Rejections - 35 USC § 103

11. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

12. Claims 1-23 and 27-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over by US PGPub 2004/0133595 to Black (hereafter Black) in view of US PGPub 2004/0002952 to Lee et al (hereafter Lee et al).

Referring to claim 1, Black discloses a computer-implemented method of selecting an abstraction level to use when generating parser output, comprising the steps of:

requesting, by an application program [browsing function] (see [0039], lines 4-6 - "the markup document 139 may be input using an appropriate browsing function"), generation of parser output [DOM 141], by a parser [PDOM parser 136] that parses an input [markup document 139] (Fig 4A, steps 203 and 206 – "Input markup document to convert to PDOM" and "Parse the markup document to produce a corresponding DOM"); and

receiving, by the application program [browsing function] (see [0039], lines 4-6 - "the markup document 139 may be input using an appropriate browsing function") from the parser [PDOM parser 136], output generated by the parser [PDOM parser 139] from the input [markup document 139], wherein the generated output adheres to a different syntax level than a syntax level (see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM

141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document).

However, Black fails to explicitly disclose the further limitation of validation of the input. Lee et al also discloses parsing an xml document which is then converted into a document object model (see abstract), including the further limitation of validating the input [xml document] (see [0023]) in order to determine if errors exist in the markup document.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the feature of Lee et al of a validating parser with the method of Black. One would have been motivated to do so in order to determine if errors exist in the markup document.

Referring to claim 2, Black/Lee discloses the method according to claim 1, wherein the validating of the input [XML validator 1113 validates errors in the XML document] (Lee et al: see [0023], lines 1-2) is performed by the parser [XML parser 111 includes an XML validator 1113] (Lee et al: see [0018], lines 5-8).

Referring to claim 3, the combination of Black and Lee et al (hereafter Black/Lee) discloses the method according to claim 1, wherein the input is a structured document [markup document 136] (Black: see Fig 4A, item 203)

Referring to claim 4, Black/Lee discloses the method according to claim 3, wherein the structured document is encoded in Extensible Markup language ("XML") (Black: see [0023], lines 1-4 – "With reference to Fig 2A, shown is an example of a markup document 139 in the form of an XML document to provide an illustration of an original document that may be represent by a corresponding DOM 141").

Referring to claim 5, Black/Lee discloses the method according to claim 1, wherein the generated output comprises at least one object representation [document object model 141] generated from the input (see Fig 4A, item 206).

Referring to claim 6, Black/Lee discloses the method according to claim 1, wherein the parser is a validating parser [XML parser 111 includes an XML validator 1113] (Lee et al: see [0018], lines 5-8 – since the parser contains a validator component, the parser is considered to represent a validating parser) that also performs validating of the input [XML validator 1113 validates errors in the XML document] (Lee et al: see [0023], lines 1-2).

Referring to claim 7, Black/Lee discloses the method according to claim 1, wherein the requesting step further comprises the step of specifying a schema name [template name 149] of a schema [template – the template defines the structure of the document and therefore is considered to represent the schema] to which the generated output must adhere (Black: see [0039], lines 8-10).

Referring to claim 8, Black/Lee discloses the method according to claim 1, wherein the requesting step further comprises the step of specifying a schema name [default template name 149] of a schema [template – the template defines the structure of the document and therefore is considered to represent the schema] to be used by the parser when generating the output (Black: see [0039], lines 8-10).

Referring to claim 9, Black/Lee discloses the method according to claim 8, wherein the schema name [default template name 149] is specified as a feature of the parser [the PDOM parser 136 generates a default template name 149] (Black: see [0039], lines 8-10).

Referring to claim 10, Black/Lee discloses the method according to claim 8, wherein the schema name [template name 149] is specified by the application program [user interface] (see [0035], lines 1-4).

Referring to claim 11, Black/Lee discloses the method according to claim 1, wherein the syntax level used for the validating of the input is specified in the input (Lee et al: see [0022] – stating which schema to use from the schema database is considered to represent inputting the syntax level).

Referring to claim 12, Black/Lee discloses the method according to claim 11, wherein the specification in the input uses a schema location construct [xsi:schemaLocation] in the input (Lee et al: see [0022]).

Referring to claim 13, Black discloses a computer-implemented method of casting objects, comprising the steps of:

an input [markup document 139] according to a first syntax level (Fig 4A, steps 203 and 206 – "Input markup document to convert to PDOM" and "Parse the markup document to produce a corresponding DOM") while generating an output objects, from the input, according to a second syntax level than a syntax level (see [0028] – the

Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document).

providing the generated output objects for use by an application program [browser] (see [0039]).

However, Black fails to explicitly disclose the further limitation of validation of the input. Lee et al also discloses parsing an xml document which is then converted into a document object model (see abstract), including the further limitation of validating the input [xml document] (see [0023]) in order to determine if errors exist in the markup document.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the feature of Lee et al of a validating parser with the method of Black. One would have been motivated to do so in order to determine if errors exist in the markup document.

Referring to claim 14, Black/Lee discloses the method according to Claim 13, wherein the second syntax level is a less-restrictive version of the first syntax level (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level becomes less-restrictive than the first syntax level).

Referring to claim 15, Black/Lee discloses the method according to Claim 13, wherein the first syntax level is a more-restrictive definition of the second syntax level (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level becomes less-restrictive than the first syntax level; therefore the first syntax level is more-restrictive than the second syntax level).

Referring to claim 16, Black/Lee discloses the method according to Claim 13, wherein the first syntax level is an extension of the second syntax level (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level has less elements than the first syntax level, and therefore the first syntax level is considered to be an extension of the second syntax level).

Referring to claim 17, Black/Lee discloses the method according to Claim 13, wherein the first syntax level represents an extension of the second syntax level (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level has less elements than the first syntax level, and therefore the first syntax level is considered to be an extension of the second syntax level).

Referring to claim 18, Black/Lee discloses the method according to Claim 13, wherein the first syntax level and the second syntax level are defined using schemas [template – the template defines the structure of document and therefore is considered to represent the schema] (Black: see [0046]).

Referring to claim 19, Black/Lee discloses the method according to Claim 18, wherein the schema that defines the first syntax level is an extension of the schema that defines the second syntax level (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level has less elements than the first syntax level, and therefore the first syntax level is considered to be an extension of the second syntax level; the templates are considered to represent the schemas that define the syntax levels).

Referring to claim 20, Black/Lee discloses the method according to claim 13, wherein the first syntax level represents a plurality of extension to the second syntax level (Black: see [0046] – the original markup document can divided into many sections with each section representing an element; the many sections are created in the form of a template; the multiple templates are considered to represent a plurality of extensions).

Referring to claim 21, the method according to claim 13, wherein the generated output objects [markup document] adhere to the second syntax level (Black: see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document).

Referring to claim 22, the method according to claim 13, wherein the input [markup document] adheres to an extended schema that defines the first syntax level (Black: see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document).

Referring to claim 23, the method according to claim 22, wherein the generated output objects [markup document] adhere to a base schema that is extended by the extended schema (Black: see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document).

Referring to claim 27, Black/Lee discloses the computer program product according to Claim 26, wherein the first syntax level is a more-restrictive version of the second syntax level (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level becomes less-restrictive than the first syntax level; therefore the first syntax level is more-restrictive than the second syntax level).

Referring to claim 28, Black/Lee discloses the computer program product according to Claim 26, wherein the first syntax level is an extension of the second syntax level (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level has less elements than the first syntax level, and therefore the first syntax level is considered to be an extension of the second syntax level).

Referring to claim 29, Black/Lee discloses the computer program product according to claim 26, wherein the first schema is defined as an extension of some intermediate schema that extends the second schema (Black: see [0028] – adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; if elements are deleted from the original markup document, then the second syntax level has less elements than the first syntax level, and therefore the first syntax level is considered to be an extension of the second syntax level; if there are two templates that have the same elements, however, the second templates also has additional elements, then the first template is considered to represent the intermediate).

Referring to claim 30, the computer program product according to claim 26, wherein the second schema is a base schema upon which one or more extensions are based, wherein the second schema is one of the extensions and is based either directly on the base schema or on an intermediate schema that extends the base schema (Black: see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document).

Referring to claim 31, Black discloses a computer-implemented method of providing parsing for clients, comprising steps of:

providing a parser that enables a client to dynamically select an abstraction level for use when generating output from the parser [selecting which elements are to be included in the template] (see [0026] and [0029], lines 9-16);

obtaining an input document [markup document 139] to be parsed for the client (see [0039], lines 4-6 –"the markup document 139 may be input using an appropriate browsing function"); and

generating output from the input document with the provided parser, for use by the client, wherein the generated output has syntax that conforms to the abstraction level that has been dynamically selected by the client and wherein the abstraction level is a refinement of the first syntax level (see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document).

However, Black fails to explicitly disclose the further limitations of the parser being a validating parser and of validating the input document with the provided validating parser, wherein the validation is performed according to a first syntax level associated with syntax specified in the input document. Lee et al also discloses parsing an xml document, which is then converted into a document object model (see abstract), including the further limitations of the parser being a validating parser [XML parser 111 includes XML validator 1113] (see [0018], lines 5-8) and of validating the input document with the provided validating parser (see [0023]), wherein the validation is performed according to a first syntax level associated with syntax specified in the input document (see [0019]-[0023] – validation is performed based on the DTD) in order to determine if errors exist in the markup document.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the feature of Lee et al of a validating parser with the method of Black. One would have been motivated to do so in order to determine if errors exist in the markup document.

13. **Claims 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over by US PGPub 2004/0002952 to Lee et al in view of US PGPub 2004/0133595 to Black .**

Referring to claim 24, Lee et al discloses a system for applying abstraction to object markup definitions, further comprising:

a validating parser [XML parser 111 includes XML validator 1113] (see [0018], lines 5-8);

first means for using the validating parser to validate an input document expressed as an object markup definition [XML document], wherein the validation is performed according to a syntax level which allows the object markup definition to be successfully validated [a DTD is chosen based on the XML document] (see [0019] - [0023]).

However Lee et al fail to explicitly disclose second means for using the validating parser to apply abstraction to the object markup definition when generating an output object, responsive to the first means, wherein the application of abstraction generates the output object according to a different syntax level which would not allow the object markup definition to be successfully validated. Black discloses an markup document as an input into a parser (see abstract), including the further limitation of second means for using the validating parser to apply abstraction to the object markup definition when generating an output object, responsive to the first means, wherein the application of abstraction generates the output object according to a different syntax level (see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document) which would not allow the object markup definition to be successfully validated (if new elements are added to the markup document in order to create the second syntax level, then the document would not be able to be successfully validated since it is more restrictive than the original document) in order to increase efficiency of the system by reducing the amount of initial coding required for altering a document.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the feature of Black of creating a document at a different syntax level with the system of Lee et al. One would have been motivated to do so in order to increase efficiency of the system by reducing the amount of initial coding required for altering a document.

Referring to claim 25, the combination of Lee et al and Black discloses the system according to claim 24, wherein the different syntax level is requested by an application program [browser] that will consume the generated output object (Black: see [0039], lines 4-6).

Referring to claim 26, Lee et al discloses a computer program product for parsing of input, the computer program product embodied on one or more computer-usable media and comprising:

computer-readable program code for validating an input document [XML document], according to a first schema, wherein the first schema defines a first syntax level that enables content in the input to be successfully validated [a DTD is chosen based on the XML document] (see [0019] - [0023]); and

However Lee et al fail to explicitly disclose computer-readable program code for generating one or more output objects according to a second schema. Upon parsing the successfully-validated content in the input, wherein the second schema defines a second syntax level that does not enable the content in the input to be successfully validated. Black discloses an markup document as an input into a parser (see abstract), including the further limitation of generating one or more output objects according to a second schema. Upon parsing the successfully-validated content in the input, wherein the second schema defines a second syntax level that does not enable the content in the input to be successfully validated (see [0028] – the Document Model allows programmers to add, modify or delete elements or content; when a document expressed as a DOM 141 is stored in non-volatile memory, the document is translated back into the markup language from which it came; adding, modifying or deleting elements is considered to represent changing the syntax level of markup document 139; therefore, output markup document is considered to be at a different syntax level than the input markup document) which would not allow the object markup definition to be successfully validated (if new elements are added to the markup document in order to create the second syntax level, then the document would not be able to be successfully validated since it is more restrictive than the original document) in order to increase efficiency of the system by reducing the amount of initial coding required for altering a document.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the feature of Black of creating a document at a different syntax level with the system of Lee et al. One would have been motivated to do so in order to increase efficiency of the system by reducing the amount of initial coding required for altering a document.

Response to Arguments

14. Applicant's arguments regarding the 35 U.S.C. 101 rejection of claims 24 and 25 filed 20 July 2006 have been fully considered but they are not persuasive.

On page 19 of the amendment, the applicants' argue the following: "As is well known, a validating parser is not "software per se," but instead is an executable program." However, an executable program is an example of software per se. Software per se is considered to represent non-statutory subject matter and furthermore the use of terminology "usable by a computer" does not render the subject matter of software per se statutory as mention above in the 35 U.S.C. 101 rejection section.

15. Applicant's arguments with respect to claims 1-31 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- US PGPub 2004/0153967 to Bender et al titled “Dynamic Creation of an Application’s XML Document Type Definition”

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kimberly Lovel whose telephone number is (571) 272-2750. The examiner can normally be reached on 8:00 - 4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Kimberly Lovel
Examiner
Art Unit 2167

26 September 2006
kml

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

29 September 2006